

1 **1. (Currently Amended)** One or more ~~tangible computer-readable~~
2 ~~media computer storage media~~ comprising a flash memory driver that is
3 executable by a computer to interface between a file system and one or more flash
4 memory media, the flash memory driver comprising:

5 flash abstraction logic that is invokable by the file system to manage flash
6 memory operations without regard to the type of the one or more flash memory
7 media; and

8 flash media logic configured to interact with different types of the flash
9 memory media;

10 wherein the flash abstraction logic invokes the flash media logic to perform
11 memory operations that are potentially performed in different ways depending on
12 the type of the flash memory media, and further wherein the flash memory driver
13 is flash memory medium agnostic, and wherein one of the flash memory
14 operations includes performing wear-leveling operations associated with the flash
15 memory medium by way of circular and continuous advancement of a write
16 pointer[[.]], and wherein the flash memory driver resides as a component within an
17 operating system of the computer.

18
19 **2. (Cancelled).**

20
21 **3. (Original)** The flash memory driver as recited in Claim 1,
22 wherein one of the flash memory operations includes maintaining data integrity of
23 the flash memory medium.
24
25

1 **4. (Original)** The flash memory driver as recited in Claim 1,
2 wherein one of the flash memory operations includes handling recovery of data
3 associated with the flash memory medium after a power-failure.

4
5 **5. (Original)** The flash memory driver as recited in Claim 1,
6 wherein one of the flash memory operations includes mapping status information
7 associated with physical sectors of the flash memory medium for use by the file
8 system.

9
10 **6. (Previously Presented)** The flash memory driver as recited in
11 Claim 1, wherein the flash medium logic is further configured to translate
12 commands received from the file system to physical sector commands for issuance
13 to the flash memory media.

14
15 **7. (Previously Presented)** The flash memory driver as recited in
16 Claim 1, wherein the flash medium logic is user programmable to read, write and
17 erase data to and from the flash memory media.

18
19 **8. (Previously Presented)** The flash memory driver as recited in
20 Claim 1, wherein the flash media logic is configured to perform error code
21 correction associated with the flash memory media.

1 **9. (Currently Amended)** A flash driver, comprising:
2 flash abstraction logic, interposed between a file system and a flash
3 memory medium, configured to:
4 (a) map a logical sector status from the file system to a physical sector
5 status of the flash memory medium; and
6 (b) maintain memory requirements associated with operating the flash
7 memory medium;
8 wherein the flash driver is located remote from the flash memory medium,
9 and wherein the memory requirements include managing wear-leveling operations
10 associated with the flash memory medium by way of circular and continuous
11 advancement of a write pointer[[]], and wherein the flash driver resides as a
12 component within an operating system of a computer.

13
14 **10. (Original)** The flash driver as recited in Claim 9, further
15 comprising a user programmable flash medium logic, configured to read, write
16 and erase data to and from the flash memory medium.

17
18 **11. (Original)** The flash driver as recited in Claim 9, further
19 comprising a user programmable flash medium logic configured to receive and
20 translate specific operational commands from the file system associated with
21 reading and writing data to the flash memory medium.

22
23 **12. (Cancelled).**
24
25

1 **13. (Original)** The flash driver as recited in Claim 9, wherein the
2 memory requirements include maintaining data integrity of the flash memory
3 medium.

4
5 **14. (Original)** The flash driver as recited in Claim 9, wherein the
6 memory requirements include handling recovery of data associated with flash
7 memory medium after a power-failure.

8
9 **15. (Original)** The flash driver as recited in Claim 9, further
10 comprising a flash medium logic, programmably configurable by a user to perform
11 error code correction associated with the flash memory medium.

12
13 **16. (Currently Amended)** A flash driver, comprising:
14 user programmable flash medium logic, configured to read, write and erase
15 data to and from a flash memory medium; and

16 flash abstraction logic, interposed between a file system and flash memory
17 medium to maintain universal requirements for the operation of the flash memory
18 medium;

19 wherein the flash memory driver is flash memory medium agnostic, and
20 wherein the universal requirements include managing wear-leveling operations
21 associated with the flash memory medium by way of circular and continuous
22 advancement of a write pointer[[]], and wherein the flash driver is defined as a
23 component within an application.
24
25

1 **17. (Original)** The flash driver as recited in Claim 16, wherein the
2 flash abstraction logic passes specific commands associated with certain types of
3 flash memory media directly to the flash medium logic for translation and
4 execution.

5
6 **18. (Original)** The flash driver as recited in Claim 16, wherein the
7 flash abstraction logic is an interface between the flash medium logic and the file
8 system.

9
10 **19. (Original)** The flash driver as recited in Claim 16, wherein the
11 universal requirements include maintaining data integrity of the flash memory
12 medium.

13
14 **20. (Cancelled).**

15
16 **21. (Original)** The flash driver as recited in Claim 16, wherein the
17 universal requirements include handling recovery after a power-failure.

18
19 **22. (Original)** The flash driver as recited in Claim 16, wherein the
20 flash medium logic comprises a set of programmable entry points that can be
21 implemented by a user to interface with the type of flash memory medium
22 selected.

1 **23. (Currently Amended)** A processing device that uses a flash
2 memory medium for storage of data, comprising:

3 a file system, configured to control data storage for the processing device;
4 flash media logic, configured to perform physical sector operations to a
5 flash memory medium based on physical sector commands, wherein the flash
6 medium logic comprises a set of programmable entry points that can be
7 implemented by a user to interface with any type of flash memory medium
8 selected; and

9 flash abstraction logic, configured to maintain flash memory requirements
10 that are necessary to operate the flash memory medium, wherein the flash memory
11 requirements include managing wear-leveling operations associated with the flash
12 memory medium by way of circular and continuous advancement of a write
13 pointer[[]], wherein the flash media logic and the flash abstraction logic reside
14 within an operating system of the processing device.

15
16 **24. (Original)** The processing device as recited in Claim 23, wherein
17 the flash abstraction logic passes physical logic commands associated with certain
18 types of flash memory medium directly to the flash memory medium logic for
19 translation and execution.

20
21 **25. (Original)** The processing device as recited in Claim 23, wherein
22 the flash abstraction logic is an interface between the flash medium logic and the
23 file system.
24
25

1 **26. (Original)** The processing device as recited in Claim 23, wherein
2 the flash memory requirements include maintaining data integrity of the flash
3 memory medium.

4
5 **27. (Cancelled).**

6
7 **28. (Original)** The processing device as recited in Claim 23, wherein
8 the flash memory requirements include handling recovery after a power-failure.

9
10 **29. (Original)** The processing device as recited in Claim 23, wherein
11 the requirements are common to a plurality of different flash memory media.

12
13 **30. (Original)** The processing device as recited in Claim 23, wherein
14 the flash medium logic comprises a set of programmable entry points that can be
15 implemented by a user to perform error code correction with the type of flash
16 memory medium used in the processing device.

17
18 **31. (Original)** The processing device as recited in Claim 23, whereby
19 the flash medium logic relieves the flash abstraction logic from performing
20 translation of the physical sector commands received from the file system.

21
22 **32. (Original)** The processing device as recited in Claim 23, wherein
23 the physical sector operations include read, write and error code correction
24 commands associated with the flash memory medium.

1 **33. (Currently Amended)** In a processing device that uses a flash
2 memory medium for storage of data, a method for driving the flash memory
3 medium, comprising:

4 managing rules associated with operating the flash memory medium in a
5 flash abstraction logic; and

6 issuing physical sector commands directly to the flash memory medium
7 from a flash medium logic;

8 wherein the method is flash memory medium agnostic, and wherein one of
9 the rules includes managing wear-leveling operations associated with the flash
10 memory medium by way of circular and continuous advancement of a write
11 pointer[[]], and wherein the method is performed by way of a component residing
12 within an operating system of the processing device.

13
14 **34. (Original)** The method as recited in Claim 33, wherein one of the
15 rules includes maintaining data integrity of the flash memory medium.

16
17 **35. (Cancelled).**

18
19 **36. (Original)** The method as recited in Claim 33, wherein one of the
20 rules includes handling recovery of the media after a power-failure.

21
22 **37. (Original)** The method as recited in Claim 33, wherein issuing
23 physical sector commands directly to the flash memory medium comprises
24 receiving read and write commands from a file system and translating them into
25 the physical sector commands.

1 **38. (Original)** The method as recited in Claim 33, further comprising
2 issuing a set of programmable entry points that can be implemented by a user to
3 perform error code correction with the type of flash memory medium used in the
4 processing device.

5
6 **39. (Original)** The method as recited in Claim 33, further comprising
7 issuing a set of programmable entry points that can be optionally selected by a
8 user to interface with the type of flash memory medium used in the processing
9 device.

10
11 **40. (Original)** The method as recited in Claim 33, further comprising
12 receiving read and write commands from a file system.

13
14 **41. (Currently Amended)** One or more tangible computer-readable
15 media comprising computer-executable instructions that, when executed, perform
16 the method as recited in claim 33.

1 **42. (Currently Amended)** A ~~tangible computer-readable medium~~
2 computer storage media for a flash driver, comprising computer-executable
3 instructions that, when executed, direct the flash driver to provide an interface
4 between a file system, selected from one of a plurality of different file systems,
5 and a flash memory medium, selected from one of a plurality of different flash
6 memory media, wherein the flash driver is located as a component within an
7 operating system and is remote from the flash memory medium, and wherein
8 wear-leveling of the flash memory medium is performed by way of circular and
9 continuous advancement of a write pointer.

10
11 **43. (Currently Amended)** A ~~tangible computer-readable medium~~
12 computer storage media for a flash driver, comprising computer-executable
13 instructions that, when executed, direct the flash driver to:

14 provide an interface between a file system, selected from one of a plurality
15 of different files systems, and a flash memory medium, selected from one of a
16 plurality of different flash memory media; and

17 manage a set of characteristics that are common to the plurality of different
18 flash memory media at a flash abstraction logic;

19 wherein the flash driver is flash memory medium agnostic, and wherein
20 wear-leveling of the flash memory medium is performed by way of circular and
21 continuous advancement of a write pointer[[]], and wherein the flash driver
22 resides as a component within an operating system.

1 **44. (Currently Amended)** A ~~tangible computer-readable medium~~
2 computer storage media for a flash driver, comprising computer-executable
3 instructions that, when executed, direct the flash driver to:

4 provide an interface between a file system, selected from one of a plurality
5 of different files systems, and a flash memory medium, selected from one of a
6 plurality of different flash memory media;

7 manage a set of characteristics that are common to the plurality of different
8 flash memory media at a flash abstraction logic; and

9 provide programmable entry points that can be optionally selected by a user
10 to interface with the type of flash memory medium selected;

11 wherein the flash driver is located as a component within an operating
12 system and is remote from the flash memory medium and the flash driver is flash
13 memory medium agnostic, and wherein wear-leveling of a flash memory medium
14 is performed by way of circular and continuous advancement of a write pointer.

1 **45. (New)** A method, comprising:
2 providing a processor-executable application, a flash driver residing as a
3 component within the processor-executable application;
4 managing rules associated with operating a flash memory medium by way
5 of the flash driver; and
6 issuing physical sector commands directly to the flash memory medium by
7 way of the flash driver, wherein the method is flash memory agnostic by virtue of
8 the flash driver.

9
10 **46. (New)** The method as recited in Claim 45, wherein one of the
11 rules includes maintaining data integrity of the flash memory medium.

12
13 **47. (New)** The method as recited in Claim 45, wherein one of the
14 rules includes handling recovery of the media after a power-failure.

15
16 **48. (New)** The method as recited in Claim 45, wherein issuing
17 physical sector commands directly to the flash memory medium comprises
18 receiving read and write commands from a file system and translating them into
19 the physical sector commands.

20
21 **49. (New)** The method as recited in Claim 45, further comprising
22 issuing a set of programmable entry points that can be implemented by a user to
23 perform error code correction in accordance with the type of flash memory
24 medium used in a processing device.
25

1 **50. (New)** The method as recited in Claim 45, further comprising
2 issuing a set of programmable entry points that can be optionally selected by a
3 user to interface with the type of flash memory medium used in a processing
4 device.

5
6 **51. (New)** The method as recited in Claim 45, further comprising
7 wear-leveling of the flash memory medium performed by way of circular and
8 continuous advancement of a write pointer.

9
10 **52. (New)** One or more computer storage media comprising
11 computer-executable instructions that, when executed, perform the method as
12 recited in claim 45.